DOCUMENT RESUME

ED 299 741 EC 211 020

AUTHOR Hoffman, Tony
TITLE How Database Management Systems Can Be Used To Evaluate Program Effectiveness in Small School Districts.
PUB DATE Feb 88
NOTE 16p.; In: Alternative Futures for Rural Special Education. Proceedings of the Annual ACRES (American Council on Rural Special Education) National Rural Special Education Conference; see EC 211 005.
PUB TYPE Reports - Descriptive (141) -- Guides - Non-Classroom Use (055) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer System Design; *Database Management Systems; *Disabilities; Elementary Secondary Education; Enrollment; *Individualized Education Programs; Information Retrieval; Information Storage; Microcomputers; Models; Program Effectiveness; *Program Evaluation; *School Districts; School District Size; Scores; Special Education
IDENTIFIERS *Small School Districts

ABSTRACT

Sophisticated database management systems (DBMS) for microcomputers are becoming increasingly easy to use, allowing small school districts to develop their own autonomous databases for tracking enrollment and student progress in special education. DBMS applications can be designed for maintenance by district personnel with little technical support. A survey of 14 California and Oregon school districts analyzed their use of DBMS for special education records. Methods that a DBMS uses to organize data are described. A strategy is presented for developing a database for local program evaluation in small districts, by using a relational DBMS system including files for Students, Enrollments, and Scores. The model system is derived from the individualized educational planning process mandated for special education programs. Such a system can be implemented on an Apple Macintosh, IBM PC, or IBM-compatible computer. Figures illustrate sample data entry forms and sample computer-generated reports. (JDD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED299741

How Database Management Systems Can Be Used to Evaluate Program Effectiveness in Small School Districts

Tony Hoffman, Ph.D.
San Lorenzo Valley Unified School District

Author's address:
1211 Laurent Street
Santa Cruz, CA 95060
(408) 429-5901

A paper presented at the eighth annual
American Council on Rural Special Education
National Rural Special Education Conference
February 24-27, 1988, in Monterey, California.

EC 211020

BEST COPY AVAILABLE

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Doris Helge
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Abstract

Sophisticated database management systems (DBMSs) for microcomputer systems are now becoming increasingly easy to use, allowing small school districts to develop their own autonomous databases for tracking enrollment and student progress in Special Education. DBMS applications can be designed that can be maintained by district personnel with little technical support. This paper presents a strategy for developing a database for local program evaluation in small districts by presenting a conceptual model of a database for program evaluation, and by displaying the use of a sample database from such a system. It is suggested that such internal evaluation systems will efficiently provide documentation that is helpful in program review and program development.

Tony Hoffman, Ph.D.
1211 Laurent Street
Santa Cruz, CA 95060

How Database Management Systems Can Be Used to Evaluate Program Effectiveness in Small School Districts

Teachers and administrators are increasingly interested in evaluating special education programs so they can provide themselves with accurate feedback and inform program reviewers about the effectiveness of their special education programs. This means that districts must keep large sets of data, called "databases," that can be analyzed to describe the progress of students in their special programs.

In the past, the expense and time required to collect and analyze database information was the privilege of large districts who could afford to employ data technicians, particularly those who had access to university mainframe computers. These districts were able to present rather impressive data analyses to program reviewers, and hence they probably had an advantage in their appeals for grants and special funding. They certainly had a good source of information that could assist in internal program review and internal program enhancement.

Developments in computer hardware and software technology in the last five years have made affordable office computer systems available that would be more than adequate in processing data for students in small to medium-sized school districts. Computers such as the Apple Macintosh™, IBM AT™, and IBM-compatibles now have quite sophisticated data processing capabilities, which not only allow complex data management, but allow elaborate capabilities for desktop publishing of data output. Current hard disk technology permits ample record storage. Networking capabilities are increasing, permitting database use by different people at different sites. A wide array of data management software, called "database management systems" (or DBMSs), is available, ranging from sophisticated networking database languages to "friendly" single-machine programs (Bryan, 1988; Custer, 1988; Finkelstein & Pascal, 1988; Greitzer, 1986; Harrington, 1987, 1988a, 1988b; Petreley, 1987; Spezzano, 1987). DBMS "applications" (i.e., DBMSs programmed to manage data in a specified manner appropriate for the specific applied needs of an organization) are relatively easy to design with some programs, while others permit quite sophisticated data management.

Depending on the application, sophistication and ease of design are both possible. Some of the easiest and least sophisticated "relational" DBMSs (DBMSs that can relate different sets of data to each other) are probably more than

adequate in managing the data needed to follow the progress of students through a small district's special education programs. The improving quality of the software is reducing the need to rely on programming consultants and data technicians. This situation permits school districts to keep their own autonomous databases.

Current DBMS Use For Special Education Recordkeeping

An informal survey conducted by the author in 1987 suggests that some California and Oregon districts are just becoming aware of these new possibilities in data management. In phone calls to twelve California districts and two Oregon districts, DBMSs for special education records were being used in eleven of the districts. Each of these districts had used "canned" enrollment programs (preset programs which had set structure and functions), nine being networked to regional offices. Seven of these districts had found that the inflexibility of the canned programs hampered their recordkeeping.

These districts are now involved with their regional offices in "upgrading" their hardware, software, and networks so they can use more flexible and powerful DBMSs (cf. Melymuka, 1986; Greitzer, 1986). Programming consultants have been contracted to implement the upgrades. The popular DBMS dBase III (now dBase III Plus) is being used in all seven cases on IBM-PC compatibles.

The new systems will allow these districts to store and analyze more data than they could previously. The systems will greatly enhance the districts' database networking capabilities, and they will also allow the districts to enhance and change the structure of their recordkeeping systems over time. Installation will involve training key personnel and the provision of instructional guides that both explain the DBMS and the districts' specific application of the DBMS. Such steps are essential to ensure that a database remains useful (Melymuka, 1986; Pepper, 1986). If training and installation are properly implemented, the districts should be less reliant on consultants than they were when they were using the less powerful and less manipulable canned programs.

It is important to note that all systems used by the districts surveyed, including the most advanced upgrades, have one purpose: to track enrollment. Districts installing DBMSs with adequate active memory and storage can do much more. Importantly, they can be used to track the academic progress of special education students through their programs. To illustrate this capability, some explanation of the terms and functions of DBMSs is needed.

How DBMSs Organize Data

DBMSs run on a simple model of data organization (the complexity of DBMSs has more to do with how data is managed). Each bit of information describing an

entity is called an "entry." DBMSs have entries organized into basic categories, which are called "fields." A collection of related fields of data pertaining to one entity is called a "record." The collection of all the data in all the records is called a "file." A file is the highest level of data organization in a DBMS; the terms "file" and "database" can be used interchangeably.

Imagine the many categories of Special Education data that describe students. Each student has a name, date of birth, handicapping condition(s), enrollment date, et cetera, all of these being categories describing the student. These categories could be fields in a database. Records would consist of the information in each field that pertains to a particular student. The entire collection of records describing all the students in the district would be a complete file. Let us call this the "Student" file.

But information can be filed in different ways, and to the extent that the records and fields in a database need to be organized differently, there may a need to file the same information in different ways. Now imagine a separate database file, called "Enrollment," that describes the characteristics of classes rather than students. Class type, teacher name, class size, et cetera would be fields describing each class. Importantly, each class would have a list of enrolled students. Hence, there is relation between the Enrollment and Student files, i.e.,. that each student is enrolled in a class. The two files have a logical "link," which allows information in one file to be used in another. A "relational" DBMS can manage the exchange and use of information between two or more files through the use of such links. This not only allows for time to be saved by avoiding redundant entries in the separate files, but it also allows the analysis of relations between student characteristics and their enrollment.

Links can define relations between files so that records in one file are related to more than one record in another. The latter file is called a "repeating collection." For example, students in Special Education are periodically tested and retested with the same tests so that a repeating collection of records in a "Scores" file could pertain to one student in the Student file. The relations between the Student file and the Enrollment file involve an even more complex type of linking, since one student can be enrolled in more than one class over time, and one class will usually have more than one student at any one time. Hence, the Enrollment and Student files would have links designed to have repeating collections of each other. The art of DBMS design involves planning and defining file structures and their links so data is related easily and meaningfully.

Tracking Academic Progress

Figure 1 is a model of a DBMS system using sample Student, Enrollment, and Scores files described above. The model is representative of visual displays that

can be used by some programs (for example, Reflex Plus, dBase Mac, and Fourth Dimension - Harrington, 1987, 1988a; Custer, 1988) to design database applications.[1] The model illustrates the three files, each with fields pertinent to describing information commonly kept in hardcopy Special Education files. Fields are also included that logically link the files to each other.

The model is derived directly from the IEP (Individualized Educational Plan) process mandated for Special Education programs. Upon initial referral, a student file is set up allowing basic characteristics of the student to be described. Subsequently, evaluation results in the inclusion of test scores and derived IEP goals. Enrollment follows, a direct result of the IEP process. Separating the Scores and Enrollment files from the initial Student file permits repeated collections of evaluation data (which consists of test results and evaluation of IEP goals) and repeated enrollment in Special Education programs.

The underlined fields in the files are called "key" fields. These are fields that uniquely define each record. Key fields are used by many DBMSs to efficiently identify individual records, and thereby speed information processing. The last name, first name, and date of birth will usually suffice to identify a key Student record (if not, a unique marker could be added on one of two identical sets of key records). Program, teacher name, and location of a Special Education class would suffice as keys for Enrollment. Review date, student name and birthdate could be used as keys in the Scores file, but a redundant entry of the names and birthdate would be required. DBMSs can be designed to insert a unique sequence number automatically to make a key; the Test# field would be used for this purpose.

Having defined each unique record, the links between files can then relate different records in different files. The evaluation data obtained on different review dates for the same student are linked as a repeating collection so that for every Student record there is a set of one or more Scores records (each pertaining to each time the student is reevaluated). One Enrollment record is linked to one or more sets of students' Scores. This latter result does not reflect that test scores or IEP goals necessarily dictate certain programs, but rather the more basic relation that a set of evaluated students will be enrolled in one program at any point in time.

The model allows the tracking of students over time by linking their repeating evaluations and their repeating enrollment in different programs in different school years. The linking structure allows for the portrayal of a

---

[1] Reflex Plus, by Borland International (4585 Scotts Valley Drive, Scotts Valley, CA, 95066), was used with an Apple Macintosh for the research in this paper. The use of Reflex Plus in this paper not meant to portray either an endorsement or a criticism of the program (see reviews by Spezzano, 1987, and Custer, 1988), nor is the use of an Apple Macintosh meant as a recommendation that other microcomputers be ignored. There are at least six DBMSs for the Apple Macintosh that would be adequate for the purposes described in this paper; there are at least 40 for IBM PCs and IBM compatibles.

student's evaluation results and enrollment history. It also allows the portrayal of the results of being enrolled in different programs. Hence, information pertinent to program evaluation and individual student progress can be accessed from the same DBMS.

The model allows much flexibility in data entry. By definition, Special Education students have highly different characteristics and needs, as a result, very different evaluations occur for different students. By planning fields in the Scores file that allow entries for scores from the many different evaluation instruments that could be used, records for virtually any Special Education student can be kept. For most DBMSs, it is not a problem to add fields to a file after the application is in use (providing file and linking structure remains intact).

Entry Forms

The design of such a DBMS application should include "entry forms" that allow easy data input. The "forms" consist of screen documents that are designed to present fields for data entry. Entry forms should be designed to make data entry as easy as possible. Fields and field labels can be moved to allow rapid sequential entry, or arranged so they resemble a hardcopy data form submitted by teachers.

The following examples utilize an entry form and database design that is abbreviated, for the purposes of illustration, to show only academic progress in the Scores file (again, adding fields to such a file should be no problem, even after data has been entered in the abbreviated form), using fabricated data. Figure 2 shows a screen display of an entry form for the file system. The fields are represented by boxes and provided with labels describing them.

Most DBMSs can be designed so that the information that is entered meets certain specifications. For example, the fields that have dates can be prespecified so that typographical errors in form (e.g., 3/15/766) will immediately be followed by a request to correct the form error. Even more strict limitations can be made; for example, if one knows that all review dates will be in the 87-88 school year, entries can be limited to only allow dates within that year. The number of decimals in a number field, such as the grade equivalent fields, can be prespecified to have only one decimal place to the right of the decimal point. Fields with letters can also be constrained: fields for programs can be constrained to only allow entries such as "RSP," "SDC-LH," and abbreviations for the other programs within a district. Entry of teacher names can be constrained to allow only their correct spellings.

DBMSs have "programming languages" that allow such constraints to be created. Many have guided menus that facilitate the derivation of equations such as "Program = RSP OR Program = SDC-LH OR . . . ." Equations can be easily

derived to search for files (as in Figure 2, where in the top panel an equation searching for "Dean" was entered) and to make "calculation fields." These are fields that automatically display the results of certain calculations in each record. For example, the average growth of a student in reading and math may be calculated in the Scores file. Class size could be automatically calculated with each entry into the Enrollment file in the Number of Students field.

Blank entries also pose little difficulty. Entries may be left blank for fields of reading and mathematics achievement for profoundly handicapped students. Since intelligence testing often does not occur in each review, blank entries for IQ scores can occur in some of a student's Scores records, and can be filled in others.

The repeating collection of the Scores file is represented by the fields surrounded by solid bars in Figure 2. Entry of the results of repeated evaluations can be included in those fields using commands that reproduce them each time the student is reevaluated and new data is to be entered. Thus, entry forms are not only used to open new records, but to edit and add information to existing records. If such editing needs to be restricted, many DBMSs provide methods so data can be protected.

Separate Entry Systems

The entry form in Figure 2 allows entry for all the information in the Student, Scores, and Enrollment files. This avoids the redundant entry of data regarding enrollment, program type, and class assignments. This is a convenience, but there may be times when redundancy may be preferred. For example, each year special programs are planned prior to enrolling existing students. When planning and implementing such programs, it would be preferable to enter their existence in a separate entry form anticipating enrollment.

Many DBMSs allow the use of separate entry forms for data entering the same filing system. Figure 3 is an example of an entry form for a class. The teacher, program year, and location were entered. Initially, the repeating collection of anticipated students was blank; as students are enrolled, the collection displays their names with selected information already available from other files. The field calculating the number of students is programmed to automatically count the number of students in the class as they are entered. Thus, an entry form can be designed to display and calculate information obtained from other files.

The additional entry form can also enter information in fields already included in the original entry form. Enrollment information in the entry form in Figure 2 can be obtained by making entries in the form in Figure 3. The file structure in Figure 1 allows entries so students with existing records can

simply be enrolled in new programs, and this new enrollment will be automatically entered into their Student and Scores records.

Reports

"Reports" can be either screen displays or printouts of data obtained from the filing system. Like entry forms, their fields can be displayed in different ways, and thus arranged to present information of interest. The information in reports can consist solely of data from an individual record or as compilations of data obtained from multiple records.

Figure 4 is an example of a report from an individual student record. Note that the data is rearranged from how it was presented in the entry form in Figure 2. Calculation fields are added showing the selected student's average growth.

Figure 5 is an example of a report listing a set of records in the Enrollment file. DBMSs provide search equations that specify which records are to be selected and the order in which they are displayed. Figure 5 is a selection of all Special Education classes for the 1987-1988 school year presented in alphabetical order.

Like entry forms, reports can use information from different files. Figure 6 is an example of a report listing a set of individual records from the Scores file presented for a certain class (which is defined by an Enrollment record). Again, calculations can be made on the selected data, allowing an analysis of class and program effectiveness.

DBMSs currently on the market vary in the extent to which calculation fields can be designed for the purposes of making reports about enrollment and student progress. The great majority of DBMS use is for business applications, so some of the statistical applications of interest to educators and program reviewers are neglected by some DBMSs. DBMSs on the current market also vary in the extent they can make graphical displays of data, such as line graphs and bar charts.

Despite these difficulties, reports with basic statistics can be obtained from most DBMSs. Such reports are what all the recordkeeping and data management is meant for. DBMSs generally make such reports easy to design, and the data they include easy to search for. Rather impressive printouts can be made quickly and efficiently. It should be hoped that such program review can be helpful in program improvement and program development.

Selecting Hardware and Software

The DBMSs that are capable of managing a file system such as in Figure 1 generally require what are referred to as "office" microcomputers. Again, these include the Apple Macintosh, IBM ATs, and IBM compatibles. They tend not to

include the Apple IICs, Apple IIEs, and Commodore 64s that are so popular as educational microcomputers. Suitable DBMSs run best with a maximum of active memory, so an IBM AT or compatible would be preferred over an IBM XT, and a Macintosh Plus or SE would be preferred over earlier Macintosh versions. Even in a small district with only two or three hundred Special Education students, the multiplicity of floppy diskettes needed to store their records would greatly hamper the speed and efficiency of a DBMS. A hard disk that contains all the student data should be used with the machine; its size should be determined from estimates of record size and student enrollment.

Record size will depend on the DBMS used and on the design of the file system. Hence, it is important to select hardware and software together. A consultant who specializes in DBMSs should probably be used for this purpose. It is fortunate that many of the DBMSs available can more than meet the needs of a small district, but caution is nevertheless warranted: be sure that suggested hardware and software works for the specific needs of the district before a purchase is made.

This may not be very difficult. In ten of the districts surveyed above, office microcomputers that could run suitable DBMSs were already owned by the districts. Hard disks for these systems were available in eight of the districts. Suitable DBMSs were already owned by three of the districts that had not upgraded their recordkeeping systems. It may simply be a matter of communicating with local staff to find the software and hardware needed to run a file system for Special Education students.

Future Developments

DBMSs in the microcomputer industry are just coming of age (Bryan, 1988, Harrington, 1988b; Finkelstein & Pascal, 1988). Many new capabilities are expected, which can only work to the advantage of organizations such as school districts who need specialized applications. Graphic and statistical capabilities in DBMSs are improving. "Friendly" DBMSs can be expected that make application design even easier. "Communications" are the big theme for 1988. Networking capabilities between offices and between file systems are increasing. Districts' abilities to share data (with respect for confidentiality laws), ideas, and evaluation systems will be greatly increased. This means only good news for small districts, since availability of information with which they can enhance their programs will be increased.

References

Bryan, M. 1988: The year of the data base. Personal Computing, January 1988, 12(1), pp. 100-109.

Custer, L. Trusting your reflexes. MacUser, February 1988, 4(2), pp. 137-145.

Finkelstein, R., & Pascal, F. SQL Database management systems. Byte, January 1988, 13(1), 111-118.

Greitzer, J. Exercising the upgrade option. In: Database management software, a supplement to PC Week, April 29, 1986, 3(17), pp. 25-35.

Harrington, J. An entirely new dimension. MacUser, November 1987, 3(11), pp. 98-106.

Harrington, J. dBase is loaded. MacUser, February 1988, 4(2), pp. 123-133. (a)

Harrington, J. Relational database management for microcomputers: Design and implementation. New York: Holt, Rinehart, &Winston, 1988. (b)

Melymuka, K. When you've outgrown your DBMS. In: Database management software, a supplement to PC Week, April 29, 1986, 3(17), pp. 1-6.

Pepper, J. Introducing new systems to users. In: Database management software, a supplement to PC Week, April 29, 1986, 3(17), pp. 15-22.

Petreley, N. Relational databases. InfoWorld, August 24-31, 9, pp. 47-53.

Spezzano, C. Database managers: A sample of the latest products for MS-DOS and the Mac. Byte, Summer 1969, 12(7), pp. 77-88.

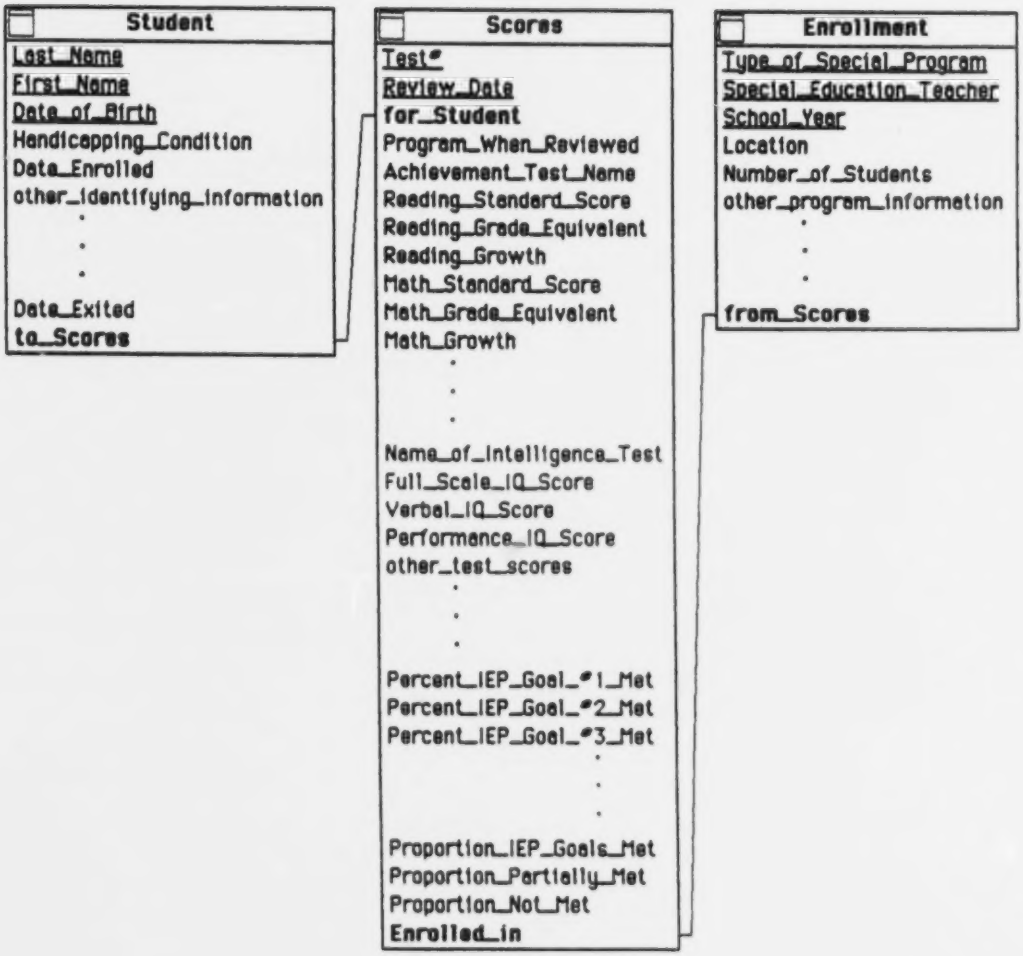

Figure 1. A model of a database design for tracking student progress and enrollment. Each box represents a separate file, with an outline of possible fields. The lines between files represent linked fields.

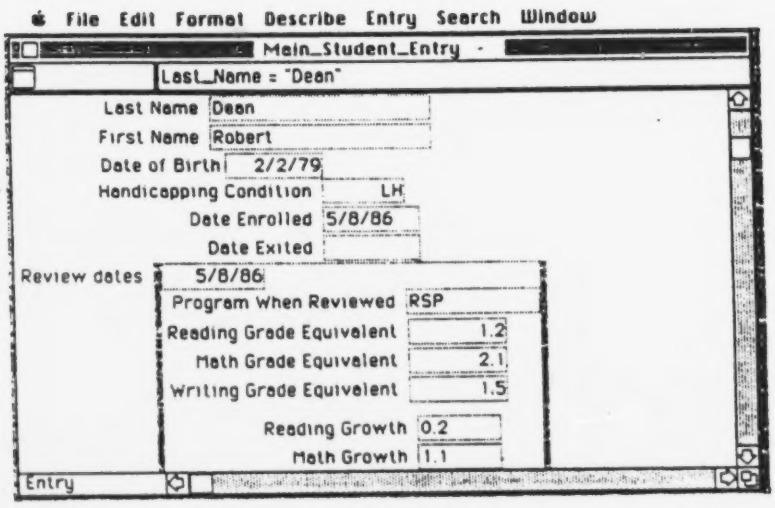

Figure 2. The screen for the beginning of an entry form, derived from the file system outlined in Figure 1, with entries for a record.

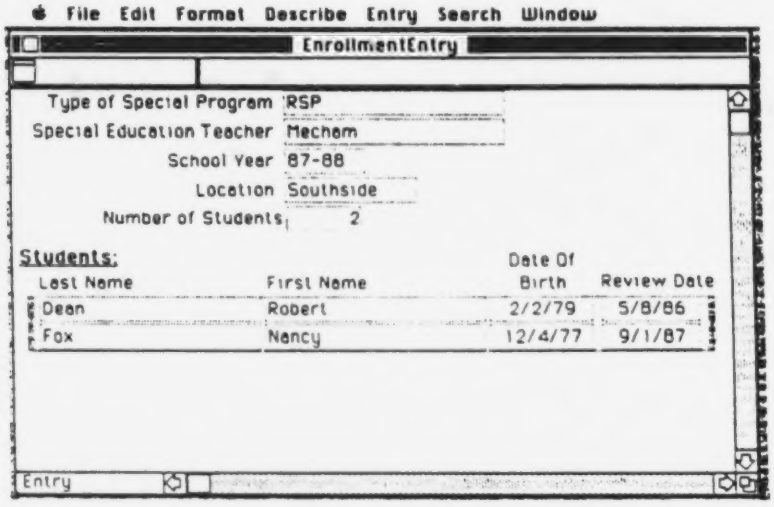

Figure 3. A separate entry form for the Enrollment file, derived from the file system outlined in Figure 1, with entries for a record for an RSP class. The Enrollment file is linked to the other files outlined in Figure 1, so that two students enrolled using the entry form in Figure 2 are now enrolled in the class.

Last Name: Jewett
First Name: Peter
Date of Birth: 5/9/72 Handicapping Condition: LH
Date Enrolled: 6/7/83 Date Exited:

Scores:	Review Date	Program When Reviewed	Reading Grade Eq	Math Grade Eq	Writing Grade Eq
	6/15/87	SDC-LH	4.9	7.4	5.8
	6/18/88	SDC-LH	5.4	8.3	6.1
	5/4/89	SDC-LH	6.1	8.7	6.5

Growth:	Review Date	Program When Reviewed	Reading Growth	Math Growth	Writing Growth
	6/15/87	SDC-LH	0.4	1.1	0.5
	6/18/88	SDC-LH	0.5	0.9	0.3
	5/4/89	SDC-LH	0.7	0.4	0.4
		Average yearly growth	0.5	0.8	0.4

Figure 4. An example of a student report, created from data entered into the entry form in Figure 2.

Jefferson Unified School District

Special Education Classes for 87-88 School Year

Program	Teacher	School Year	Location
RSP	Johnson	87-88	Kennedy HS
RSP	Jones	87-88	Northside
RSP	King	87-88	MLKJr. High
RSP	Mecham	87-88	Southside
SDC-LH	Martinsen	87-88	Kennedy HS
SDC-LH	Norman	87-88	Southside
SDC-LH	Smith	87-88	Northside
SDC-SED	Spooner	87-88	Southside
SDC-SH	Miller	87-88	Northside

Figure 5. A list of classes and programs, derived and selected from sample data entered into the entry forms in Figure 2 and Figure 3

Academic Growth in Mr. Norman's LH-SDC
Southside Elementary School

Program	Reading Growth	Math Growth	Writing Growth
SDC-LH	1.1	1.1	0.8
SDC-LH	0.2	1.1	0.7
SDC-LH	0.6	0.8	0.4
SDC-LH	0.7	0.4	0.4
SDC-LH	0.5	0.9	0.3
SDC-LH	0.4	1.1	0.5
SDC-LH	1.0	1.3	0.8
SDC-LH	1.1	0.7	1.1
SDC-LH	0.6	0.4	0.8
SDC-LH	0.5	0.8	0.2
SDC-LH	1.0	1.1	0.9
SDC-LH	0.7	0.2	0.8
SDC-LH	1.1	1.3	1.2
Average Growth	0.7	0.9	0.7
Standard deviation	0.3	0.4	0.3

Figure 6. A report of students' academic growth over a one year period in a Special Day Class, derived from data entered into the entry forms in Figures 2 and 4, with calculation fields providing summary statistics.